**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**
**IMPACT FACTOR: 7.056**

***IJCSMC, Vol. 14, Issue. 2, February 2025, pg.32 – 43***

# Indonesian Gold Price Prediction: A Machine Learning Approach Using Random Forest Regressor

**Hawangga Dhiyaul Fadly[1]; Fatchul Arifin[2]**

[1]Departement of Electronics and Informatics Engineering Education
Universitas Negeri Yogyakarta, Yogyakarta, Indonesia
[2]Departement of Electronics and Informatics Engineering Education
Universitas Negeri Yogyakarta, Yogyakarta, Indonesia
[1] hawanggadhiyaul.2023@student.uny.ac.id; [2] fatchul@uny.ac.id

**DOI: https://doi.org/10.47760/ijcsmc.2025.v14i02.004**

**Abstract: This study applies the Random Forest Regressor algorithm to predict the price of Indonesian gold, using economic factors such as the Indonesia Composite Index (IHSG), silver prices (Perak), IDR/USD exchange rate, and crude oil prices (Minyak_Mentah). The dataset, spanning from 2015 to 2020 with 1,333 data points, is used for training and testing the model. The R-squared value achieved by the model is 0.9804, indicating that approximately 98% of the variance in gold prices is explained by the selected variables. The results show that the IDR/USD exchange rate has the highest correlation with gold prices (0.634892), followed by silver prices (0.546647), IHSG (0.224411), and crude oil prices (-0.027445), highlighting the significant role of currency and economic factors in gold price fluctuations. While the Random Forest model provided accurate predictions, the study acknowledges limitations, particularly the limited number of commodities considered. Further research should include additional variables and compare other machine learning algorithms, such as Gradient Boosting or Neural Networks, to improve prediction accuracy. The findings emphasize the value of machine learning in financial forecasting, offering a useful tool for investors and market analysts in predicting gold price trends.**

**Keywords**: **Gold Prediction, Random Forest Regressor, Machine Learning**

## I. INTRODUCTION

Gold, a precious metal with inherent value, has long been utilized not only as a medium of exchange but also as a store of wealth and a key asset in investment portfolios[1][2]. The distinct characteristics of gold are attributed to its production processes, which occur through magmatic cycling and fixation on the Earth's surface. These processes contribute to its rarity, making gold one of the most coveted commodities[3][4]. As a fine, malleable metal that resists corrosion, gold is not only valued for its economic properties but also for its utility in the creation of aesthetically pleasing jewelry and valuable items. Its use as a currency and form of wealth dates back to ancient civilizations, providing a deep historical connection to human commerce.

Gold is often regarded as one of the most stable forms of investment[5]. Its intrinsic value tends to remain relatively steady, making it an attractive option for investors seeking to preserve wealth. Investment refers to the allocation of capital to assets or projects with the expectation of earning returns or generating profit over time. Investors may turn to gold as a preferred investment option, especially in times of economic uncertainty, as it is perceived to retain its value better than many other assets[6][7]. This characteristic makes gold an essential hedge against inflation, currency devaluation, and other risks associated with market volatility. Gold's ability to preserve purchasing power, even in turbulent times, contributes to its reputation as a "safe haven" investment, providing security against adverse economic shifts[8][9][10].

Given the complexities of predicting gold prices, investors often rely on predictive analytics to make informed decisions. Forecasting gold prices accurately is vital, as it enables investors to mitigate risks and capitalize on favorable market conditions[11][12]. Gold prices are subject to a wide array of influences, including macroeconomic factors such as interest rates, inflation expectations, and geopolitical uncertainty. By understanding and predicting these factors, investors can enhance their ability to make timely and effective investment choices. Predicting gold prices is a challenging task due to the volatility of financial markets and the multitude of external variables involved. To address this challenge, various predictive models have been developed, including statistical techniques and machine learning algorithms. Among these, the Random Forest Regressor algorithm has emerged as a powerful tool for predicting asset prices, including gold[13]. Random Forest, a supervised ensemble learning method, builds multiple decision trees and aggregates their predictions to improve accuracy and robustness. This algorithm is well-suited for forecasting gold prices because it can capture complex, nonlinear relationships between the input variables and the target price, which are often difficult for traditional linear models to handle[13][14].

The Random Forest Regressor operates by first splitting the dataset into random subsets, training a separate decision tree on each subset, and then averaging the predictions of all trees[15][16][17]. Each tree in the forest makes a decision based on different features of the data, and the final output is the average prediction from all trees. This process helps reduce overfitting, which is common in individual decision trees, and enhances the model's ability to generalize across unseen data. In the context of gold price prediction, relevant features might include historical gold prices, economic indicators such as inflation rates and interest rates, commodity prices (such as oil), and financial data reflecting global market sentiment[18]. By training the model on these features, the Random Forest algorithm can learn the underlying patterns that drive gold price fluctuations, enabling it to make predictions about future price movements. Additionally, the Random Forest model is known for its interpretability, allowing analysts to assess the importance of different features in determining the price of gold. For example, the model might reveal that oil prices and the US dollar have a more significant impact on gold prices than other variables, providing valuable insights for investors looking to optimize their portfolios[19]. The ability to analyze feature importance is particularly beneficial in financial markets, where understanding the drivers of price movements is crucial for making strategic decisions.

In practice, the Random Forest Regressor can be applied to historical price data to predict future trends in gold prices[20][21]. By leveraging the power of machine learning and big data, investors can refine their strategies and make more data-driven decisions. The algorithm's predictive capabilities, combined with its ability to account for complex market dynamics, make it a promising tool for forecasting the often-unpredictable behavior of gold prices. Furthermore, Random Forest models can be continuously updated with new data, allowing for real-time predictions and more accurate decision-making in an ever-changing market environment. Ultimately, the integration of machine learning models such as the Random Forest Regressor into gold price prediction strategies offers a powerful tool for investors. These models provide more precise forecasts, improve risk management, and help investors capitalize on market opportunities. By incorporating these predictive techniques into investment decisions, individuals and institutions can make more informed choices, optimize their portfolios, and better navigate the volatility of the global financial market.

## II. METHODOLOGY

This study uses secondary data from trusted financial sources, including https://id.investing.com and https://www.bullion-rates.com, covering gold, silver, the rupiah exchange rate, crude oil prices, and the Composite Stock Price Index (CSPI) from 2015 to 2020. The dataset, which contains 1,333 data points, was chosen to include variables known to influence gold prices. These commodities, particularly silver and crude oil, are closely linked to market conditions and the global economy, while the rupiah exchange rate and CSPI are key indicators of economic stability, making them essential in predicting gold prices.

The Random Forest algorithm is ideal for handling large, complex datasets with multiple features, making it suitable for forecasting gold prices, which are influenced by various economic factors. The algorithm works by training multiple decision trees on random subsets of data, combining their predictions to improve accuracy. This ensemble approach enhances the model’s robustness and reliability, offering a more precise prediction than individual decision trees[22][23]. The following figure is the stages of the random forest regressor:

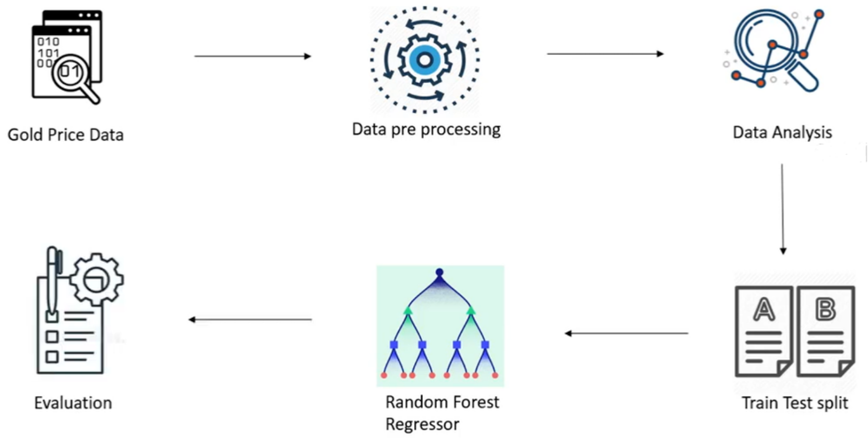


Fig. 1 The stages of random forest regressor

The Random Forest Regressor method for predicting gold prices involves several stages, each contributing to the development of an accurate and reliable model. These stages include data collection, preprocessing, analysis, model training, prediction, and evaluation.

A. *Data Collection*

This study begins with the collection of historical data on gold prices and other influential commodities spanning from 2015 to 2020. The dataset comprises key variables such as gold prices, silver prices, crude oil prices, the rupiah exchange rate against the US dollar, and the Composite Stock Price Index (CSPI), which are selected due to their significant impact on gold price fluctuations. The use of high-quality, reliable data ensures the accuracy and robustness of the predictive model.

B. *Data Preprocessing*

Preprocessing is conducted to prepare the dataset for analysis and modeling. This phase involves addressing missing values through imputation techniques or the removal of incomplete records, depending on the extent of data gaps[24]. Categorical variables are transformed into numerical formats using encoding techniques such as one-hot encoding or label encoding to meet the input requirements of the Random Forest Regressor algorithm. Furthermore, normalization or standardization is applied to ensure that all variables are on a comparable scale, thereby enhancing model performance and mitigating potential biases.

C. *Data Analysis*

The preprocessed data undergo exploratory data analysis (EDA) to understand its statistical properties and underlying patterns. Descriptive statistics, including measures of central tendency (mean, median), dispersion (standard deviation, variance), and range (minimum, maximum, quartiles), are calculated to identify potential outliers and anomalies[25]. Additionally, correlation analysis is performed to assess the strength and direction of relationships between gold prices and other commodities, such as silver, crude oil, and the rupiah exchange rate. This analysis aids in feature selection, allowing the model to focus on variables that significantly influence gold price movements

D. *Train-Test Spilt*

To evaluate the model's generalization capability, the dataset is partitioned into training and testing subsets using an 80:20 ratio. The training set (80% of the data) is utilized to develop and optimize the Random Forest Regressor model, while the testing set (20% of the data) serves to assess the model's predictive performance on unseen data. This approach helps prevent overfitting and ensures that the model's predictions are reliable when applied to real-world scenarios.

E. *Random Forest Regression*

The Random Forest Regressor algorithm, an ensemble learning technique based on multiple decision trees, is implemented to predict gold prices. Each decision tree is trained on a random subset of the training data, and the final prediction is derived by averaging the outputs of all individual trees. This method enhances the model's accuracy and robustness by reducing the variance associated with single decision trees and capturing complex, non-linear relationships between gold prices and other explanatory variables

F. *Model Evaluation*

The performance of the Random Forest Regressor model is evaluated using several statistical metrics to assess its predictive accuracy. The coefficient of determination (R-squared) is employed to measure the proportion of variance in gold prices explained by the model[26]. Additionally, error metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are calculated to quantify the model's prediction errorsa[27]. Based on these evaluation results, the model may undergo further refinement through hyperparameter tuning or the inclusion of additional relevant features to improve its predictive performance.

## III.RESULT AND DISCUSSION

This research focuses on analyzing precious metal prices, with a particular emphasis on gold, utilizing the Google Colab platform. Google Colab provides a cloud-based environment that supports Python programming and allows for easy integration with machine learning libraries, making it an ideal tool for testing and evaluating the Random Forest Regressor model. Here are the steps involved in testing the Random Forest Regressor algorithm using Google Colab:

### A. *Importing Required Libraries*

The first step is to import the necessary libraries for data manipulation, analysis, and visualization. Each library serves a distinct purpose in the workflow, and their functionalities are outlined below:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
```

Fig. 2 Importing required libraries

*1) Import numpy as np*

Numpy is used for array manipulation, data handling, and performing mathematical computations that support the analysis of the dataset

*2) Import pandas as pd*

Pandas is used to load the dataset, manage data, and perform preprocessing tasks like cleaning and organizing the data into a suitable format for the machine learning model.

*3) Import matplotlib.pyplot as plt*

Matplotlib is utilized to visualize trends, distributions, and correlations within the dataset, helping to better understand the relationships between the various features and the target variable (gold prices).

*4) Import seaborn as sns*

Seaborn is used for more sophisticated data visualizations, such as correlation matrices and distribution plots, to uncover hidden patterns and relationships in the data.

*5) From sklearn.model_selection import train_test_split*

This function from the Scikit-learn library is used to split the dataset into training and testing subsets. Typically, 80% of the data is used for training the model, while the remaining 20% is reserved for testing. This ensures that the model is validated on unseen data, which helps assess its ability to generalize and predict future gold prices.

*6) From sklearn.ensemble import RandomForestRegressor*

In this research, it is used to predict the price of gold based on various commodities that influence its value.

*7) From sklearn import metrics*

Scikit-learn's metrics module provides a collection of performance evaluation metrics for machine learning models, such as accuracy, R-squared, mean squared error (MSE), and root mean squared error (RMSE). These metrics are essential for evaluating the performance of the Random Forest model, allowing researchers to understand how well the model predicts gold prices and identify areas for improvement

### B. *Dataset Import and Processing*

The next step in the analysis is to import the prepared dataset and perform initial checks to ensure the data is loaded correctly. This involves using the pandas library to read the CSV file containing the data. The dataset consists of five commodities—Indonesia Composite Index (ICI), gold, silver, IDR/USD exchange rate, and crude oil prices—spanning a total of 1,333 data points. The dataset is loaded into a Pandas DataFrame, which allows for easy manipulation, analysis, and visualization of the data. Once the dataset is imported, the first task is to display the first five and the last five rows of the data to provide a quick overview of its structure. This can be accomplished using the head() and tail() functions in Pandas. These functions display the first and last five rows, respectively, allowing for an initial inspection of the data, including checking for any missing or erroneous values. In addition to viewing a sample of the data, the next step is to check the dimensions of the dataset. The shape attribute in Pandas is used to determine the number of rows and columns in the DataFrame. This gives an overview of the dataset's size and confirms that it contains 1,333 rows and 6 columns, which is essential for ensuring the completeness of the data.

 

```
[ ] # loading the csv data to a Pandas DataFrame
    gold_data = pd.read_csv('/content/dataset harga emas indonesia.csv')
```

```
[ ] # print first 5 rows in the dataframe
    gold_data.head()
```

| | Tanggal | IHSG | Emas | Perak | IDR/USD | Minyak_Mentah |
|---|---|---|---|---|---|---|
| 0 | 2/1/2015 | 5242.77 | 478114.0 | 6347.01 | 12542.5 | 660864.33 |
| 1 | 5/1/2015 | 5219.99 | 489501.0 | 6576.78 | 12627.5 | 631880.10 |
| 2 | 6/1/2015 | 5169.06 | 496694.0 | 6743.27 | 12657.5 | 606673.98 |
| 3 | 7/1/2015 | 5207.12 | 495950.0 | 6767.45 | 12738.5 | 619728.03 |
| 4 | 8/1/2015 | 5211.83 | 491648.0 | 6656.78 | 12680.0 | 618657.20 |

```
[ ] # print last 5 rows of the dataframe
    gold_data.tail()
```

| | Tanggal | IHSG | Emas | Perak | IDR/USD | Minyak_Mentah |
|---|---|---|---|---|---|---|
| 1328 | 24/06/2020 | 4964.73 | 801128.0 | 7957.72 | 14130.0 | 537081.30 |
| 1329 | 25/06/2020 | 4896.73 | 801409.0 | 8086.58 | 14175.0 | 548856.00 |
| 1330 | 26/06/2020 | 4904.09 | 807641.0 | 8110.56 | 14220.0 | 547327.80 |
| 1331 | 29/06/2020 | 4901.82 | 808483.0 | 8145.73 | 14245.0 | 565526.50 |
| 1332 | 30/06/2020 | 4905.39 | 818574.0 | 8368.50 | 14255.0 | 559793.85 |

```
[ ] # number of rows and columns
    gold_data.shape

    (1333, 6)
```

Fig. 3 Dataset import and processing

C. *Display Information and Check for Missing Data*

Once the dataset is imported, the next step is to examine its structure and check for any issues, such as missing values, that may affect the analysis. This is done using the info() function in Pandas, which provides basic information about the dataset, including the number of entries, data types of each column, and whether any missing values exist. The info() function reveals that the dataset consists of 1,333 entries, and it shows that each column has no missing or null values, as all columns are marked as "non-null." The Tanggal column, representing the date, is of object data type, while the remaining columns—IHSG, Emas, Perak, IDR/USD, and Minyak_Mentah—are all of type float64. This indicates that the date is stored as a string or categorical variable, while the other columns are numerical and contain continuous data, which is expected for financial variables. Additionally, to further verify that there are no missing values in the dataset, the isnull().sum() function is used. This function checks each column for null or missing values and sums the number of missing entries. The results confirm that there are no missing values in any of the columns, which is a positive outcome as missing data can often lead to inaccurate analysis and modeling.

```
[ ] # getting some basic informations about the data
    gold_data.info()

    <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 1333 entries, 0 to 1332
    Data columns (total 6 columns):
     #   Column         Non-Null Count  Dtype
    ---  ------         --------------  -----
     0   Tanggal        1333 non-null   object
     1   IHSG           1333 non-null   float64
     2   Emas           1333 non-null   float64
     3   Perak          1333 non-null   float64
     4   IDR/USD        1333 non-null   float64
     5   Minyak_Mentah  1333 non-null   float64
    dtypes: float64(5), object(1)
    memory usage: 62.6+ KB
```

```
[ ] # checking the number of missing values
    gold_data.isnull().sum()

    Tanggal          0
    IHSG             0
    Emas             0
    Perak            0
    IDR/USD          0
    Minyak_Mentah    0
    dtype: int64
```

Fig. 4 Display Information and Check for Missing Data

D. *Data Display the Statistical Summary of the Dataset*

The next step is to display the statistical measures of the dataset, which provide a summary of the central tendency, dispersion, and shape of the distribution of the data. This can be done using the describe() function in Pandas, which generates the following statistical measures for each numeric column in the dataset:

1) Count: The number of non-null entries in the dataset. This value should match the total number of rows in the dataset (1,333 in this case), indicating that there are no missing values for the column.
2) Mean: The average value of the dataset for each column. This represents the central value of the dataset and helps identify the overall trend of the data. For example, the mean of the Emas (gold) column indicates the average gold price during the period under study.
3) Standard Deviation (std): This measures the dispersion or spread of the data around the mean. A higher standard deviation indicates that the values are more spread out, while a lower standard deviation suggests that the values are closer to the mean. For instance, the Emas column has a standard deviation of 80461.556589, suggesting significant variation in gold prices.
4) Min: The minimum value in the dataset. This provides the lowest data point recorded for each commodity. For example, the minimum value for Emas (gold) indicates the lowest gold price observed during the time period.
5) 25% (First Quartile, Q1): This value represents the first quartile, or the point that divides the lowest 25% of the data from the rest. It gives an indication of the lower range of the dataset. For example, the Emas column's 25% value of 52288.4090 shows the price level below which 25% of the gold prices fall.
6) 50% (Median, Second Quartile, Q2): The median represents the middle value of the dataset, or the second quartile. It is the value that divides the dataset into two equal halves. The median is less sensitive to extreme values compared to the mean and provides a more robust measure of central tendency. For instance, the Emas column’s 50% value of 56080.0000 is the middle point for gold prices.
7) 75% (Third Quartile, Q3): This value represents the third quartile, or the point that divides the highest 25% of the data from the rest. It provides insight into the upper range of the dataset. For the Emas column, the 75% value of 59364.0000 shows the gold price level above which 25% of the prices are observed.
8) Max: The maximum value in the dataset. This gives the highest data point recorded for each commodity. For example, the maximum value for Emas (gold) shows the highest price of gold observed during the study period.

```
[ ] # getting the statistical measures of the data
    gold_data.describe()
```

| | IHSG | Emas | Perak | IDR/USD | Minyak_Mentah |
|---|---|---|---|---|---|
| count | 1333.000000 | 1333.000000 | 1333.000000 | 1333.000000 | 1.333000e+03 |
| mean | 5572.268777 | 576715.776444 | 7239.293226 | 13768.545386 | 7.118154e+05 |
| std | 644.557975 | 80461.556859 | 514.357135 | 634.455873 | 1.590955e+05 |
| min | 3937.630000 | 456243.000000 | 5826.110000 | 12472.500000 | 1.547045e+05 |
| 25% | 5042.870000 | 522808.000000 | 6878.800000 | 13320.000000 | 6.168268e+05 |
| 50% | 5702.820000 | 560080.000000 | 7224.300000 | 13644.500000 | 6.989822e+05 |
| 75% | 6117.360000 | 593864.000000 | 7516.020000 | 14137.500000 | 8.049369e+05 |
| max | 6689.290000 | 873369.000000 | 8907.930000 | 16575.000000 | 1.151881e+06 |

Fig. 5 Display the Statistical Summary of the Dataset

E. *Display the Correlation Between Commodities and Gold*

The value of gold is significantly influenced by the prices of other commodities. Understanding the correlation between gold and these commodities is essential for identifying the factors that drive fluctuations in gold prices. To measure this relationship, we calculate the correlation coefficient between gold prices and the other commodities included in the dataset, such as the Indonesia Composite Index (IHSG), silver prices (Perak), the rupiah exchange rate against the US dollar (IDR/USD), and crude oil prices (Minyak_Mentah). The correlation coefficient is a statistical measure that describes the strength and direction of the relationship between two variables. The values range from -1 to 1, where:

1) 1 indicates a perfect positive correlation (both variables move in the same direction),
2) -1 indicates a perfect negative correlation (one variable increases while the other decreases),
3) 0 indicates no correlation (no predictable relationship between the variables).

 

```
[ ] # correlation values of GLD
    print(correlation['Emas'])

    IHSG             0.224411
    Emas             1.000000
    Perak            0.546647
    IDR/USD          0.634892
    Minyak_Mentah   -0.027445
    Name: Emas, dtype: float64
```

Fig. 6 Display the Correlation Between Commodities and Gold

Based on the correlation results presented above, the highest correlation with gold prices is observed with the rupiah exchange rate against the US dollar (IDR/USD), with a correlation coefficient of 0.634892. This suggests a moderate to strong positive correlation, meaning that as the value of the rupiah fluctuates against the US dollar, gold prices tend to move in the same direction. On the other hand, the lowest correlation is observed with crude oil prices (Minyak_Mentah), with a very weak negative correlation of -0.027445. This indicates that changes in crude oil prices have almost no influence on gold prices.

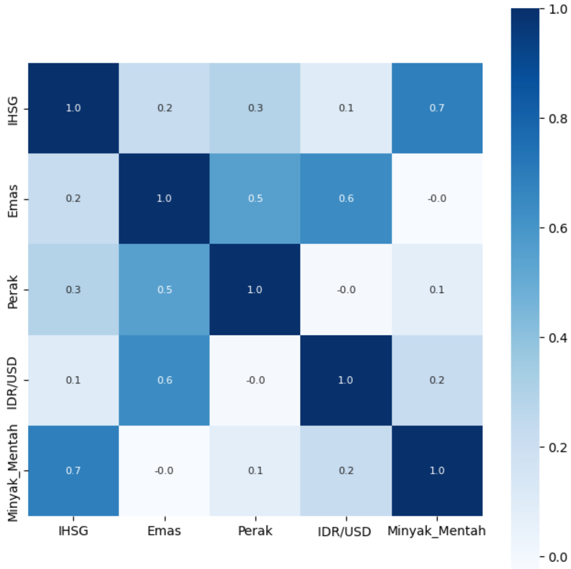


Fig. 7 Correlation Heatmap Between Commodities and Gold

Additionally, a heatmap is used to visually represent the correlation matrix between all the commodities and gold. A heatmap is a graphical representation where each cell in the matrix is colored according to the value of the correlation coefficient. The darker the color, the stronger the correlation, either positive or negative. Lighter colors represent weaker correlations, with white or light blue shades typically indicating near-zero correlations. From the heatmap, we can observe the overall relationships between the commodities. The stronger correlations (depicted by darker shades) suggest more significant influences on gold prices, while weaker correlations (represented by lighter shades) suggest less influence. This visual tool is valuable for quickly identifying which variables have the most impact on gold price movements.

F. *Display the Distribution of Gold Prices*

To gain insights into the distribution of gold prices over the period of the dataset, a histogram is created to visually represent the frequency of gold prices within specific ranges. The histogram shows how many times each price range (bin) occurs within the dataset. Based on the histogram of the 1,333 data points, it can be observed that the highest concentration of gold prices is in the range of Rp500,000 to Rp600,000. The histogram reveals that the majority of the gold prices fall within this price range, indicating that it is the most common price level over the observed period.

 

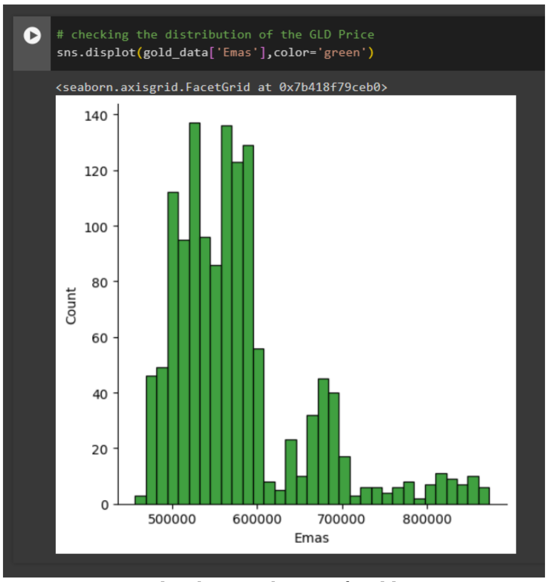

Fig. 8 Display the Distribution of Gold Prices

The resulting plot visually shows the frequency of different gold price values, with the Emas column representing the gold prices. The taller bars in the histogram represent the price ranges with higher frequency, while shorter bars indicate less common price ranges. From the histogram, we can conclude that the majority of gold prices during this period were concentrated in the Rp500,000 to Rp600,000 range, with a noticeable decline in frequency as the price increases beyond this range. This distribution provides valuable insight into how gold prices behaved during the period under study.

G. *Separating Data into X and Y Variables*

The feature variables include the Indonesia Composite Index (IHSG), silver prices (Perak), IDR/USD exchange rate, and crude oil prices (Minyak_Mentah). These are stored in variable X. The target variable, which is the gold price (Emas), is stored in variable Y. This separation is done by selecting the appropriate columns from the dataset. The drop() function in Pandas is used to remove the Tanggal (date) and Emas (gold) columns from X, as these are not needed for predicting gold prices. The remaining columns are the predictors that will help train the model. The gold price column (Emas) is then stored separately in Y, as it is the target we aim to predict.

```
[ ] X = gold_data.drop(['Tanggal ','Emas'],axis=1)
    Y = gold_data['Emas']
```

```
[ ] print(X)

          IHSG    Perak  IDR/USD   Minyak_Mentah
0      5242.77  6347.01  12542.5       660864.33
1      5219.99  6576.78  12627.5       631880.10
2      5169.06  6743.27  12657.5       606673.98
3      5207.12  6767.45  12738.5       619728.03
4      5211.83  6656.78  12680.0       618657.20
...        ...      ...      ...             ...
1328   4964.73  7957.72  14130.0       537081.30
1329   4896.73  8086.58  14175.0       548856.00
1330   4904.09  8110.56  14220.0       547327.80
1331   4901.82  8145.73  14245.0       565526.50
1332   4905.39  8368.50  14255.0       559793.85

[1333 rows x 4 columns]
```

```
[ ] print(Y)

0       478114.0
1       489501.0
2       496694.0
3       495950.0
4       491648.0
          ...
1328    801128.0
1329    801409.0
1330    807641.0
1331    808483.0
1332    818574.0
Name: Emas, Length: 1333, dtype: float64
```
Fig. 9 Separating Data into X and Y Variables

H. *Divide the Data into Training and Testing Sets*

The next critical step in the machine learning process is to split the dataset into two subsets: one for training the model and one for testing its performance. This separation ensures that the model can be evaluated on data that it has not seen during the training phase, providing a better assessment of its ability to generalize to new, unseen data. In this case, the data is divided into X (the feature variables) and Y (the target variable, gold prices). We use an 80% training and 20% testing split, which means that 80% of the data will be used to train the model, and the remaining 20% will be used to test the model's performance. This is a common split ratio in machine learning to ensure that the model has enough data to learn from while still being able to evaluate its performance on unseen data.

```
[ ] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state=2)
```

Model Training: Random Forest Regressor

```
[ ] regressor = RandomForestRegressor(n_estimators=100)
```

```
[ ] # training the model
    regressor.fit(X_train,Y_train)

    ▾ RandomForestRegressor
    RandomForestRegressor()
```

Fig. 10 Divide the Data into Training and Testing Sets

Once the data is divided into training and testing sets, we proceed to initialize and train the model. In this case, we use the Random Forest Regressor algorithm, which is an ensemble method that creates multiple decision trees and combines their predictions to make a final output. This approach helps reduce overfitting and improves the model's accuracy. We initialize the Random Forest Regressor with n_estimators=100, which means the algorithm will build 100 decision trees during the training process. This is typically a good starting point, as a higher number of estimators can help improve model performance, though it may also increase computation time. After training, the model is ready to make predictions on the test data (X_test) to evaluate its performance and predict future gold prices.

I. *Data Evaluation Model*

The goal of this evaluation is to assess how well the trained model can predict the price of gold based on the features (JCI, silver, IDR/USD, and crude oil prices) that were used during training. To do this, we use the predict() method of the trained Random Forest Regressor model to make predictions on the test dataset (X_test). This method applies the learned patterns from the training data to the unseen test data and produces predicted values for gold prices.

```
[ ] # prediction on Test Data
    test_data_prediction = regressor.predict(X_test)
```

```
print(test_data_prediction)
```

```
[558116.59 859247.45 554669.32 547380.43 517515.97 609206.49 588250.4
 664596.3  529569.09 492233.06 539410.92 587886.35 501428.36 576370.56
 594244.34 590565.71 679953.63 579312.59 509357.21 608701.23 583389.04
 508700.82 536937.6  556991.47 500355.43 533238.86 586858.8  590706.27
 595887.67 586350.64 555206.12 544154.26 595204.39 558050.4  587810.61
 563218.26 583929.64 530995.11 476884.54 645565.29 586228.34 558810.33
 840820.26 542687.76 491119.2  584974.82 559674.42 576280.46 531447.48
 599766.23 533281.18 541444.13 545004.49 588637.71 526400.3  558820.3
 615561.33 586272.94 533320.21 580359.63 576183.41 775063.26 567117.85
 587110.26 589822.23 506017.99 585520.   571243.13 716738.49 589699.34
 562565.6  640020.6  580963.59 597863.4  558451.7  856160.78 492397.3
 558378.35 496071.63 534425.47 677181.42 529177.93 694808.97 659320.27
 609443.59 527118.8  472491.92 582615.36 585745.42 600650.6  583434.05
 675154.41 511155.9  563475.69 589590.11 518882.64 685641.18 504496.12
 663925.46 505539.6  476888.07 584973.14 584364.06 580554.65 776809.04
 664199.7  527161.77 699828.57 541031.45 519320.65 640408.51 560059.
 845641.3  583603.48 584842.08 674677.93 609304.8  561156.45 539410.02
 540680.46 608330.11 826828.71 597807.2  535099.32 494985.27 494475.82
 589742.77 472395.41 587102.58 506175.64 561131.99 597302.51 472015.64
 539941.   519553.16 591824.45 572083.97 490545.35 561301.13 535037.41
 629578.16 494405.95 538696.31 522304.24 761303.94 542082.48 500660.
 521387.97 515849.28 536045.07 689946.36 534021.73 561055.56 689997.38
 679323.39 596778.09 592570.28 592362.1  523283.57 692173.24 791381.12
 580417.78 474307.91 531962.75 708733.89 716979.35 494613.46 580626.91
 502853.02 595756.08 543004.66 513539.56 509218.39 589865.85 565835.91
 574155.09 529940.97 499144.08 675047.12 523629.55 672184.96 533538.85
 815979.79 523588.93 482323.44 571516.16 587237.85 502907.35 617656.89
 482525.78 537039.24 542673.58 544824.65 741395.92 530434.07 520551.07
 499933.82 651083.46 842136.45 597497.47 623788.84 505635.11 589214.68
 675433.17 689153.01 524574.06 569489.94 577968.87 528916.23 523128.72
 520759.09 705452.01 595439.42 533636.04 675048.44 528842.22 497733.29
 753716.7  504383.57 525209.42 665170.12 591654.98 505283.98 500677.8
 498923.15 546071.61 680295.7  473146.58 508144.84 578187.97 520185.42
 475667.53 535606.85 503166.8  545669.46 521178.25 526768.79 502501.45
 540340.33 858019.01 520303.4  496594.33 562712.03 529999.33 663579.93
 684892.35 752691.29 496889.31 503069.7  544876.08 596730.04 583672.85
 584307.65 506960.86 509198.74 655945.33 514846.27 495781.68 600224.47
 593872.72 599213.   482506.7  525522.1  561527.04 562942.15 528537.29
 564168.3 ]
```

Fig. 11 Data Evaluation Model

These predictions represent the model's forecast for the gold price based on the patterns it has learned during the training phase. By comparing these predicted values with the actual gold prices (Y_test), we can assess the model's accuracy and identify areas for improvement. The printed output shows the predicted values of gold prices for the test data, which can then be further evaluated using various metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), or R-Squared to quantify the model's performance and accuracy.

J. *Predict R-Squared Error*

R-Squared is a statistical measure that indicates how well the independent variables (the features) explain the variation in the dependent variable (the target). Specifically, it provides the proportion of the variance in the target variable (gold prices) that is predictable from the feature variables (JCI, Crude Oil, Silver, and IDR/USD exchange rate). An R-Squared value ranges from 0 to 1, where:

1) 0 indicates that the model does not explain any of the variance in the target variable (gold prices),
2) 1 indicates that the model perfectly explains all of the variance in the target variable.

In this study, the R-Squared value is calculated using the metrics.r2_score() function from Scikit-learn, which compares the actual values (Y_test) to the predicted values (test_data_prediction). The formula calculates the ratio of the explained variance to the total variance, and the result is a measure of how well the model has captured the underlying patterns in the data.

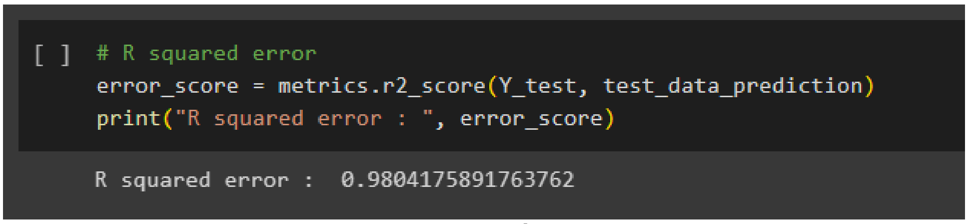


Fig. 12 R-Squared Error

K. *Prediction Visualization*

To evaluate the performance of the Random Forest Regressor model, the actual and predicted gold prices are visualized using a line plot. This allows us to compare how closely the model's predictions align with the true values. In this visualization, matplotlib.pyplot, which was imported earlier in the script, is used to create the plot. The actual gold prices are represented by a blue line, while the predicted gold prices are shown by a green line.

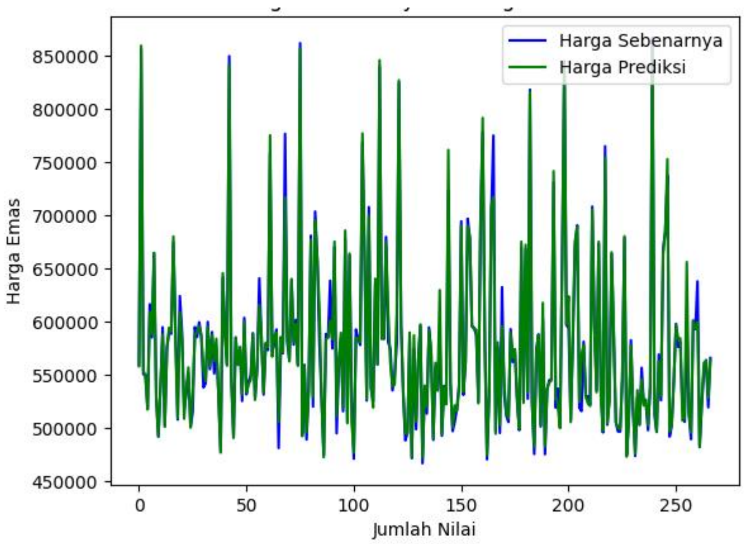


Fig. 13 prediction Visualization

The performance of the Random Forest Regressor was evaluated using the R-Squared ($R^2$) error metric, with the model achieving an impressive value of 0.9804175891763762. This indicates that approximately 98% of the variance in gold prices can be explained by the selected feature variables, demonstrating the high accuracy of the model in predicting gold prices. Such a high R-Squared value reinforces the model's effectiveness in capturing the underlying patterns in the data.

 

## IV. CONCLUSION AND FUTURE WORKS

This study demonstrated the efficacy of the Random Forest Regressor algorithm in predicting the price of Indonesian gold, achieving a high R-squared value of 0.9804, indicating that the model can explain approximately 98% of the variance in gold prices. The model's predictive accuracy highlights the strength of ensemble learning methods in capturing complex, nonlinear relationships between economic variables and commodity prices. The research also identified key economic factors that influence gold price fluctuations. Among these, the rupiah exchange rate against the US dollar (IDR/USD) exhibited the strongest correlation with gold prices (0.634892), followed by silver prices (0.546647). The Indonesia Composite Index (JCI) and crude oil prices showed weaker correlations, suggesting that while stock market performance and oil prices impact gold prices, their effects are less pronounced. These findings underline the importance of currency fluctuations and precious metal markets in shaping the price of gold, particularly in Indonesia.

However, the study acknowledges several limitations. The selection of commodities was restricted to a few key variables, which may not fully capture the range of economic influences on gold prices. Future research should consider expanding the feature set to include additional variables such as interest rates, global inflation, and geopolitical factors. Furthermore, comparing the performance of the Random Forest model with other machine learning algorithms, such as Gradient Boosting Machines (GBM) or Neural Networks, could offer valuable insights into the most effective methods for forecasting gold prices. In conclusion, while the Random Forest Regressor algorithm provides a robust and accurate tool for predicting gold prices, further exploration of additional economic variables and alternative algorithms could enhance the model's accuracy and reliability. This research contributes to the growing body of knowledge on machine learning applications in financial forecasting, offering practical implications for investors and policymakers in the gold market.

### ACKNOWLEDGEMENT

This research was fully supported by a discussion group at the Department of Electronics and Informatics Engineering Education, Universitas Negeri Yogyakarta, Indonesia

# REFERENCES

[1] N. Moji, B. Matlou, S. Mncela, F. Koning, and S. C. Shongwe, "A Statistical Analysis to Evaluate whether Gold Is Still the Safe Haven of Old," *J. Stat. Appl. Probab.*, vol. 13, no. 5, pp. 1447–1463, 2024, doi: 10.18576/jsap/130503.

[2] P. Kowalewski and D. A. Skopiec, "Price processes in the global gold market," *Bank i Kredyt*, vol. 55, no. 4, pp. 381–424, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85201203766&partnerID=40&md5=b7a3cfe7bd4aad80d78fb4a9aee41034.

[3] K.-M. Wang and Y.-M. Lee, "Could gold serve as an exchange rate hedge in Japan?," *Eng. Econ.*, no. 2, pp. 160–170, 2010, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-77954063763&partnerID=40&md5=2bdf400a189fdfe3e8ad5601d33cd820.

[4] R. Abreu and C. Pinho, "The Importance of Gold in the Financial Report," in *CSR, Sustainability, Ethics and Governance*, 2018, pp. 83–95.

[5] S. Fatima, C. Gan, and B. Hu, "Price Stability Properties and Volatility Analysis of Precious Metals: An ICSS Algorithm Approach," *J. Risk Financ. Manag.*, vol. 15, no. 10, 2022, doi: 10.3390/jrfm15100465.

[6] V. R. Dondeti, C. B. McGowan, and B. B. Mohanty, "Two perennial golden dilemmas: Should gold be a part of an investor's portfolio? Should the U.S. return to the gold standard?," *J. Int. Bus. Econ.*, vol. 15, no. 3, pp. 53–68, 2015, doi: 10.18374/JIBE-15-3.4.

[7] M. F. Ghazali, N. F. M. Ussdek, H. H. Lean, and J. W. Taunson, "Does Gold Still Shelter Inflation, and, if so, When? Evidence From Four Countries," *Asia-Pacific J. Oper. Res.*, vol. 39, no. 4, 2022, doi: 10.1142/S0217595922400036.

[8] D. Bredin, T. Conlon, and V. Potì, "Does gold glitter in the long-run? Gold as a hedge and safe haven across time and investment horizon," *Int. Rev. Financ. Anal.*, vol. 41, pp. 320–328, 2015, doi: 10.1016/j.irfa.2015.01.010.

[9] H. Boubaker, J. Cunado, L. A. Gil-Alana, and R. Gupta, "Global crises and gold as a safe haven: Evidence from over seven and a half centuries of data," *Phys. A Stat. Mech. its Appl.*, vol. 540, 2020, doi: 10.1016/j.physa.2019.123093.

[10] B. Będowska-Sójka and A. Kliber, "Is there one safe-haven for various turbulences? The evidence from gold, Bitcoin and Ether," *North Am. J. Econ. Financ.*, vol. 56, 2021, doi: 10.1016/j.najef.2021.101390.

[11] D. Makala and Z. Li, "Prediction of gold price with ARIMA and SVM," in *Journal of Physics: Conference Series*, 2021, vol. 1767, no. 1, doi: 10.1088/1742-6596/1767/1/012022.

[12] Y. Wang and T. Lin, "A Novel Deterministic Probabilistic Forecasting Framework for Gold Price with a New Pandemic Index Based on Quantile Regression Deep Learning and Multi-Objective Optimization," *Mathematics*,

 

vol. 12, no. 1, 2024, doi: 10.3390/math12010029.

[13] R. Mahajan, P. Patil, D. Chikmurge, and S. Barve, “Forecasting Gold Price using Ensemble based Machine Learning Approach,” 2023, doi: 10.1109/ICSES60034.2023.10465360.

[14] D. H. Badr, “Artificial intelligence techniques recruitment for gold ore (Crude gold) price prediction,” *Edelweiss Appl. Sci. Technol.*, vol. 8, no. 6, pp. 1044–1056, 2024, doi: 10.55214/25768484.v8i6.2208.

[15] G. Biau, “Analysis of a random forests model,” *J. Mach. Learn. Res.*, vol. 13, pp. 1063–1095, 2012.

[16] S. S. Patra, O. P. Jena, G. Kumar, S. Pramanik, C. Misra, and K. N. Singh, “Random Forest Algorithm in Imbalance Genomics Classification,” in *Data Analytics in Bioinformatics: A Machine Learning Perspective*, 2021, pp. 173–190.

[17] S. Bhattacharya and S. Chakraborty, “PREDICTION OF RESPONSES IN A CNC MILLING OPERATION USING RANDOM FOREST REGRESSOR,” *Facta Univ. Ser. Mech. Eng.*, vol. 21, no. 4, pp. 685–700, 2023, doi: 10.22190/FUME210728071B.

[18] C. Pierdzioch and M. Risse, “Forecasting precious metal returns with multivariate random forests,” *Empir. Econ.*, vol. 58, no. 3, pp. 1167–1184, 2020, doi: 10.1007/s00181-018-1558-9.

[19] S. Jayashankar and R. Sridaran, “Moving word cloud from visual towards text analysis to endow eLearning,” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 3493–3498.

[20] E. H. A. Rady, H. Fawzy, and A. M. A. Fattah, “Time series forecasting using tree based methods,” *J. Stat. Appl. Probab.*, vol. 10, no. 1, pp. 229–244, 2021, doi: 10.18576/JSAP/100121.

[21] S. Jayendrakamesh, T. F. Michaelraj, and L. C. Yong, “Comparing the accuracy of gold price prediction using linear regression and random forest regression algorithms for investment,” in *AIP Conference Proceedings*, 2024, vol. 3161, no. 1, doi: 10.1063/5.0229231.

[22] T. Zhu, “Analysis on the applicability of the random forest,” in *Journal of Physics: Conference Series*, 2020, vol. 1607, no. 1, doi: 10.1088/1742-6596/1607/1/012123.

[23] H. Wang, “Research on the Application of Random Forest-based Feature Selection Algorithm in Data Mining Experiments,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 10, pp. 505–518, 2023, doi: 10.14569/IJACSA.2023.0141054.

[24] A. F. Amiri, A. Chouder, H. Oudira, S. Silvestre, and S. Kichou, “Improving Photovoltaic Power Prediction: Insights through Computational Modeling and Feature Selection,” *Energies* , vol. 17, no. 13, 2024, doi: 10.3390/en17133078.

[25] S. K. Singla, R. D. Garg, and O. P. Dubey, “Ensemble machine learning methods for spatiooral data analysis of plant and ratoon sugarcane,” *Intell. Data Anal.*, vol. 25, no. 5, pp. 1291–1322, 2021, doi: 10.3233/IDA-205302.

[26] Akçapinar, Coşgun, and Altun, “Prediction of perceived disorientation in online learning environment with random forest regression,” in *EDM 2011 - Proceedings of the 4th International Conference on Educational Data Mining*, 2011, pp. 259–263, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84857466749&partnerID=40&md5=b45ceac82da9d729c77fa4eb34ff7be4.

[27] M. D. Adewale *et al.*, “Comparative Performance Evaluation of Random Forest, Extreme Gradient Boosting and Linear Regression Algorithms Using Nigeria’s Gross Domestic Products,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2024, vol. 541 LNICST, pp. 131–150, doi: 10.1007/978-3-031-51849-2_9.

 